April 2004

Experiments in Multirobot Air-Ground Coordination

Luiz Chaimowicz
University of Pennsylvania, chaimo@grasp.cis.upenn.edu

Ben Grocholsky
University of Pennsylvania, bpg@grasp.cis.upenn.edu

James F. Keller
University of Pennsylvania, jfkeller@grasp.cis.upenn.edu

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Camillo J. Taylor
University of Pennsylvania, cjtaylor@cis.upenn.edu

Experiments in Multirobot Air-Ground Coordination

Abstract
This paper addresses the problem of coordinating aerial and ground vehicles in tasks that involve exploration, identification of targets and maintaining a connected communication network. We focus on the problem of localizing vehicles in urban environments where GPS signals are often unreliable or unavailable. We first describe our multi-robot tesbed and control software used to coordinate ground and aerial vehicles. We present the results of experiments in air-ground localization analyzing three complementary approaches to determining the positions of vehicles on the ground. We show that the coordination of aerial vehicles with ground vehicles is necessary to get accurate estimates of the state of the system.

Comments

Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Experiments in Multirobot Air-Ground Coordination

Luiz Chaimowicz, Ben Grocholsky, James F. Keller, Vijay Kumar, Camillo J. Taylor
GRASP Laboratory - University of Pennsylvania
Philadelphia, PA, 19104
Email: {chaimo, bpg, jfkeller, kumar, cjtaylor}@grasp.cis.upenn.edu

***Abstract*—This paper addresses the problem of coordinating aerial and ground vehicles in tasks that involve exploration, identification of targets and maintaining a connected communication network. We focus on the problem of localizing vehicles in urban environments where GPS signals are often unreliable or unavailable. We first describe our multi-robot tesbed and the control software used to coordinate ground and aerial vehicles. We present the results of experiments in air-ground localization analyzing three complementary approaches to determining the positions of vehicles on the ground. We show that the coordination of aerial vehicles with ground vehicles is necessary to get accurate estimates of the state of the system.**

I. INTRODUCTION

The use of air and ground robotic vehicles working in cooperation has become an important goal in both military and non-military domains. Several military operations include reconnaissance and exploration of cluttered urban environments where communication and GPS localization can be unreliable. In this type of mission, groups of unmanned air vehicles (UAVs) could significatively help the ground vehicles (UGVs) by providing localization data and acting as communication relays. Alternatively, with respect to nonmilitary applications, teams of heterogeneous platforms hold the promise to provide researchers with an affordable comprehensive system for environmental mapping and monitoring of remote, inaccessible areas.

In both domains, heterogeneous multi-robot teams must be able to perform tasks in a coordinated manner. Examples include (i) air-ground behaviors that allow UAV information to guide UGV motion planning and control and (ii) cooperative localization and formation control to maximize the intelligence acquired by the robotic team. Air-ground coordination in multi-robot teams is a promising research field and few works [1]–[5] have tackled this problem so far.

More specifically, our air-ground coordination scenario includes a remote reconnaissance mission where a robotic team composed of ground and air vehicles will be dispatched to autonomously map an urban area and provide real-time data and image telemetry to a remote human operator that will be able to task robots to move to certain positions and find objects of interest. A key requirement in this type of scenario is the ability of the individual robots to localize themselves both in absolute and relative coordinates. Unfortunately, GPS reception can not be considered reliable at all times, mainly in highly cluttered three-dimensional environments. Moreover, other classical means of self-localization such as dead-reckoning from known positions through odometry or precise inertial measurement should be considered impractical as either too inaccurate or too expensive. Thus, in order for robotic vehicles to be successful, they must be capable of fusing a diverse assortment of sensed data to provide analytic redundancy to their position sensing capability. This requirement extends not only to the ability of a vehicle to maintain awareness of its own location but also the locations of at least some of its teammates, since in the terrain in which such teams will operate, loss of onboard position sensing will be routinely encountered, even if only for transient periods, by individual vehicles.

In this paper, we show some initial results in this direction. We first describe our multi-robot tesbed, composed of a blimp and a group of ground robots, and the control software used to coordinate them. We present results of experiments in air-ground localization analyzing three complementary approaches for determining the positions of vehicles on the ground: GPS, air-ground localization based on known ground features and air-ground localization based on UAV sensors. We discuss and compare the applicability and accuracy of these approaches in localizing both stationary and moving targets and argue that the coordination of aerial and ground vehicles is necessary to get accurate estimates of the state of the system.

This paper is organized as follows: Sections II and III present our hardware and software platforms. Section IV discuss the two types of position estimators used in this paper (GPS and camera). In Section V we show our experiments and discuss the results. Finally, Section VI brings the conclusion and directions for future work.

II. TESTBED

The GRASP Lab has a variety of autonomous moving platforms that enable a wide range of research activities with respect to coordination and control of UAV and UGV systems.

We have developed several generations of autonomous ground vehicles using a medium-sized radio controlled truck chassis as a base platform. The current generation features a high resolution stereo camera on a pan mount to provide 360 field of view sensing. These vehicles are configured with a Pentium 3 based laptop computer that runs the control and sensing algorithms. Sixteen cm. diameter wheels, four-wheel steering, and an independent suspension afford these vehicles a significant level of mobility in outdoor environments. On smooth surfaces they are capable of speeds up to $5m/s$. In addition to vision sensing, the vehicles are equipped with wheel odometers, WAAS (wide area augmentation system) enabled

GPS and a 3-axis inertial measurement unit with magnetic heading and dynamic attitude sensing capability. The resultant integrated platform is a vehicle that can autonomously navigate through moderately varying terrain while maintaining position and orientation sensing quality sufficient to allow images from the vehicle to be used to localize spatial features and create mosaic maps from the integrated set of images. The onboard command and control software allows the user to dynamically specify path or destination, while being able to control the onboard camera, including browsing its image database. We currently have a fleet of 5 of these vehicles.

We also have a medium sized blimp (9 meter length) that has nearly a 3 kg payload for research equipment. At present it is fully autonomous and capable of transmitting digital images back to an operator that can be used to generate maps and localize ground vehicles in its field of regard. To achieve this, it features the same GPS and inertial measurement unit as our ground vehicles, so it is capable of sensing rates and attitudes, including heading. It also features a video camera that can be slewed in azimuth and elevation and the onboard computing and communication hardware to be autonomous but also capable of being dynamically redirected by a remote human operator. While the blimp cannot be considered practical for tactical applications, it affords us a safe, inexpensive aerial platform that could readily be replaced with any configuration capable of low speed flight, including hover. More detail on this vehicle and its application to our Multiple Autonomous Robot Software project may be found at the GRASP Lab website: *http://www.cis.upenn.edu/mars/*.

The ground robot and the blimp are depicted in figure 1. In summary, both of these platforms have features that lend themselves to this type of research. These include:

1) A highly capable onboard CPU and control system software architecture to manage motion/navigation while also processing and transmitting digital images between autonomous vehicle platforms.
2) Provision for team level closed loop control laws involving both air and ground vehicles to enable them to cooperatively accomplish tasks.
3) Capability for sensor fusion of position, orientation and visual imagery data into a common database.

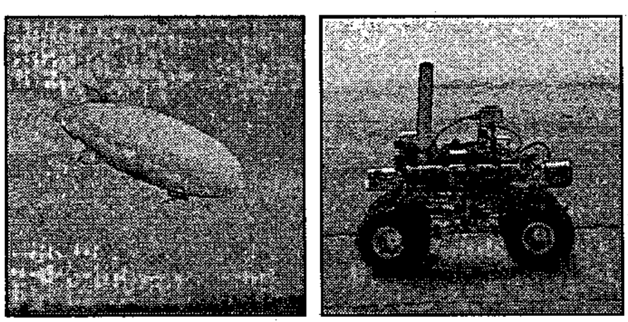


Fig. 1. GRASP air and ground vehicles.

III. Robot Software

We are using ROCI (Remote Object Control Interface) [6] for programming both the ground vehicles and the blimp. ROCI is an objected oriented programming framework that facilitates the development of applications for dynamic multi-robot teams. In this framework, each robot is considered a node which contains several processing and sensing modules and may export different types of services and capabilities to other nodes. Each node runs a kernel that mediates the interactions of the robots in a team. This kernel keeps an updated database of all nodes and the functionalities that they export. ROCI also has a browser that allows users to monitor and control tasks and nodes over the network. Users can start and stop tasks, view and log data, and change parameters of any module running in any node. Basically, a ROCI module encapsulates a process which acts on data available on the module's inputs and presents its results as outputs. They are self-contained and reusable, thus, complex tasks can be built by connecting inputs and outputs of specific modules. These connections are specified in a XML file and made through a pin architecture that provides a strongly typed, network transparent communication framework .

A. Examples

We have developed several modules and tasks for robot sensing and control using ROCI. Here we present two of these tasks: way-point navigation and leader-follower.

Autonomous navigation to designated way-points is a basic skill required for executing practical robot missions. A ROCI module has been implemented to accomplish this task by combining path following control with visual obstacle detection through ground-plane stereo. An obstacle free path is fitted through desired goal points and adjusted in the event of new obstacle detection. Linear and angular velocity commands are generated based on the track errors and obstacle proximity as depicted in figure 2. Importantly, both the path following control and registration of obstacle depend on accurate estimation of the robot absolute position and orientation.

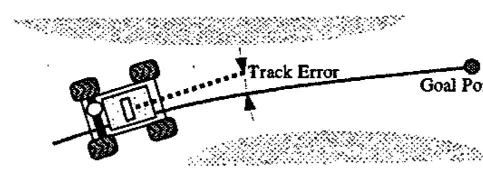


Fig. 2. Path-follower controller.

Another key skill in multi-robot teams is the ability of moving while maintaining a certain formation. One typical approach is to use a leader-follower controller, in which one of the robots (the leader) has some plan or control algorithm that moves it to a goal and one or more robots act as followers and try to keep a certain formation relative to the leader. We implemented a distributed leader-follower controller that tries to keep the follower within a desired distance (both in x and y) from the leader. The controller generates linear and angular velocities for the follower based on its distance and bearing to the leader as shown in (1). A diagram is depicted in Fig. 3.

$$\begin{aligned} v &= k_1(d\,cos(\psi) - x_{des} + y_{des}\,d\,sin(\psi) - y_{des}^2/x_{des}) \\ \omega &= k_2(d\,sin(\psi) - d/x_{des}) \end{aligned} \tag{1}$$

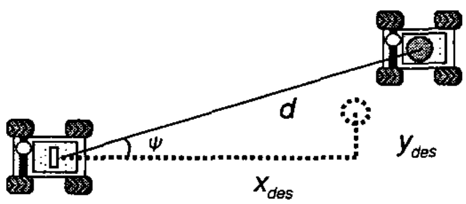


Fig. 3. Leader-follower controller.

In order to obtain d and ψ, we put a color fiducial on the leader and run a color blob extractor algorithm on the follower to segment the color and obtain its position in each image. Then, doing simple stereo processing and using information from the camera pan mount, it is possible to estimate both the distance and bearing to the leader. Figure 4 shows the several modules that compose this task. This high level of modularity allows, for example, this same controller to be used in other robots or a different sensor solution to be applied for computing the range and bearing without performing any changes on the other modules. This is one of the key features of the ROCI framework.

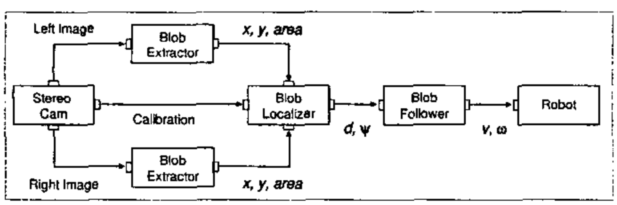


Fig. 4. Diagram of the ROCI modules for the leader-follower task.

IV. Position Estimators

A. GPS

GPS undoubtedly provides a most convenient mechanism for robot localization. However, GPS suffers afflictions that render it unsuitable as an exclusive localization sensor. Primarily, GPS is subject to varied accuracy and availability. Satellite signals may be degraded by occlusions, subject to electromagnetic interference from communications, processing and power electronic devices or corrupted by multipath reflections. Additionally, reception of WAAS measurement corrections can prove unreliable for antennas located at ground level.

Data collected during experiments conducted at the Ft. Benning MOUT site illustrate some problems encountered with GPS localization in urban environments. Figure 5 displays the UGV trajectory along with GPS position estimates and information available from the GPS receiver regarding solution quality.

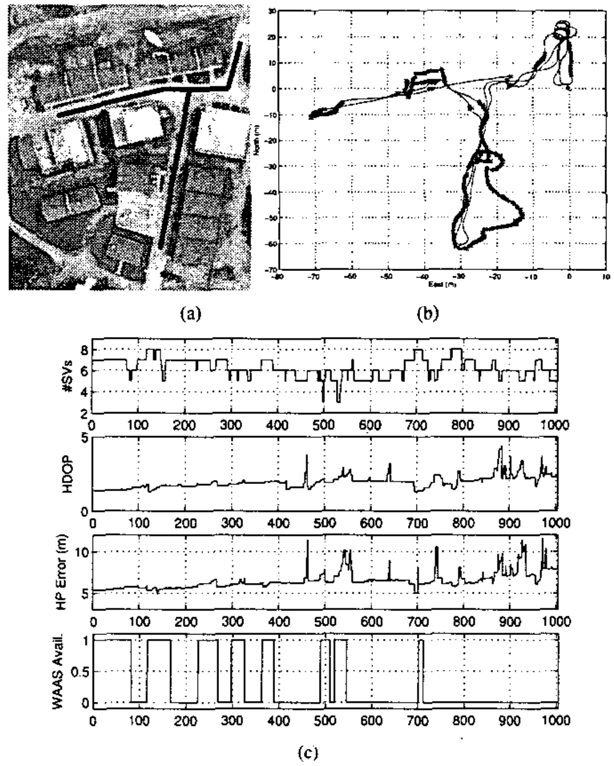


Fig. 5. GPS performance during MOUT site navigation. The UGV follows the black path indicated in aerial image (a). (b) Displays GPS position estimates. (c) Shows solution quality indicators available from the receiver (Number of Satellites, Horizontal Dilution of Precision, Horizontal Positioning Error and WAAS availability). Instances where HDOP > 2.0 are marked with 'x' in (b).

Proximity to buildings makes this an adverse environment for GPS reception. Significant localization errors are not unexpected. Importantly, as indicated in figure 5(c), areas where poor solution quality is encountered are indicated by the GPS unit itself. The limited WAAS availability experienced motivates use of conventional DGPS corrections.

Other sensors and sources of information potentially provide more precise relative and absolute position measurements than GPS. Should desired localization accuracy exceed that available from GPS, fusing additional information sources warrants the increased processing, communication and storage requirements. In situations where estimate quality from GPS alone is sufficient, resources can be released from localization to conduct mission specific sensing tasks.

B. Camera

The blimp camera can act as "an eye in the sky" and help significatively in the localization of the ground robots. Depending on how much information is available (camera calibration or a certain number features with known world coordinates) it is possible to estimate the localization of the ground robots or the pose of the blimp solely from the images. In general, given a certain image, the relation between the 3D coordinates (X^w, Y^w, Z^w) of a point in world and the 2D coordinates (x, y) of its projection on the image plane can be described by a projection matrix M, that depends on the camera's intrinsic (focus and center) and extrinsic (translation and rotation) parameters.

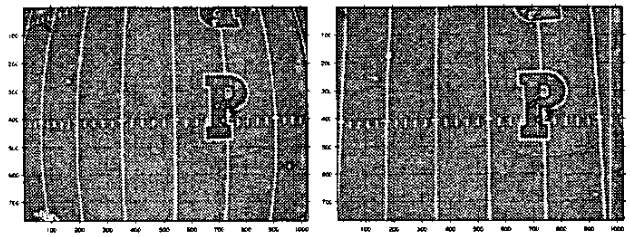
There are different ways of obtaining the matrix M. For example, from a set of correspondent image-world pairs, it is possible to estimate M using least squares techniques. These are well known computer vision equations that can be found in [7] for example. Also, knowing the translation and rotation of the camera from the blimp's GPS and IMU and the camera's intrinsic parameters it is possible to compute the projection matrix. Once M is computed, it is very simple to obtain the position of points on the ground from their projection on the image plane.

Unfortunately, both approaches have some drawbacks. When you have a fixed camera, you can compute M once from 6 known world-image pairs (less if the intrinsic parameters or other constraints are known) and then use it for localizing ground features on the image. But in our case, the pose of the camera varies constantly as the blimp moves. Consequently, M needs to be recomputed for every image but there are no guarantees that the blimp will be able to see 6 landmarks every time to compute M. On the other hand, constructing M solely based on the camera movement may not be feasible since the accuracy of the blimp's GPS and IMU may not be sufficient for localizing features within a desired error.

In the experiments presented here, all the image processing was done offline. We calibrated the camera intrinsic parameters beforehand and flew the blimp over a football field with several known markers and robots while acquiring images, GPS and IMU data. Two initial steps were necessary for the offline processing: the first one is to remove image distortions. Since our camera has a large field of view, the images from the blimp suffer from distortions that are more evident on the periphery of the image, as shown in Figure 6(a). Using a simple radial model it is possible to remove the distortions and obtain the image shown in figure 6(b). The second step is to select several pairs of world-image coordinates in order to compute M. As mentioned, we did this offline but possible approaches for online processing include placing some artificial landmarks (color blobs for example) in known world locations, which can be easily detected from the camera or having distinct features such as corners of buildings or street intersections acting as landmarks.

V. Results

In order to have accurate ground truth measurements with which to interpret all results, we conducted our experiments over the University of Pennsylvania football stadium. This afforded us a grid-like environment whose recorded dimensions could be counted on for accuracy and precision. We deployed the blimp over the field using tethers and then photographed landmarks and ground vehicles in a variety of static and dynamic positions in order to ascertain how the aggregate of sensors could be used to localize individual features or vehicles. Throughout the course of the experiment, the blimp and the ground robots logged their local position and orientation via their onboard sensors. The blimp video camera was aligned with the vehicle body axes, so that tolerances in its articulated mount did not affect results.



(a) (b)

Fig. 6. Image from the blimp before and after removal of radial distortions.

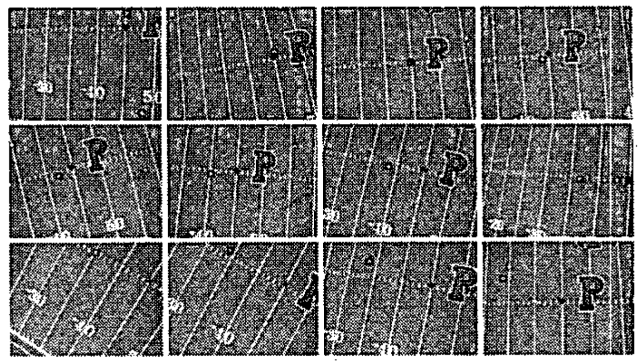


Fig. 7. Sequence of 12 images taken from the blimp with 2 second intervals. The robot position (square) and a fixed point (cross) are highlighted.

Figure 7 shows a sequence of twelve images captured from the blimp with 2 second intervals. The black squares track the positions of a moving ground vehicle and the black crosses indicate the position of a fixed landmark. Note that, in spite of being tethered, the blimp's translations and rotations change the pose of the camera from frame to frame resulting in very different views in a 24 seconds time span. We set the origin of a common world frame (W) on the 30 yard hash mark, with X being the forward direction of the field, and Z pointing up. We also considered that all points on the ground has $Z^w = 0$. To get the X^w and Y^w coordinates of a point from its GPS latitude and longitude, it is necessary to convert from degrees to meters considering the earth curvature and the latitude and longitude of the origin. Since we do not have a survey point with known latitude and longitude, we had to measure the origin coordinates using the GPS. This procedure increased the errors of all GPS measurements since we had to use this estimated origin for converting these measurements to the world frame.

A. Sensing Errors

To have an estimation of the localization errors, we compared the GPS with the air-ground localization for a fixed point with known world coordinates and computed the errors.

For the GPS we used 800 data points, after collecting data for 5 minutes and trimming the initial and final data points. For the air-ground localization we used twelve images taken from the blimp camera each 2 seconds while it was flying over the field (Figure 7). The two approaches explained in section IV-B were used for computing the projection matrix M (known world points and blimp's GPS/IMU). Since we do not have a precise measurement of the blimp's height, we estimated that it was $18m$ high throughout the experiment. For computing the localization error, we measured the distance from the estimated positions to the real position of the fixed point (13.716, 0) and computed the mean. The graph of Figure 8 shows the positions estimated and Table I summarize the errors.

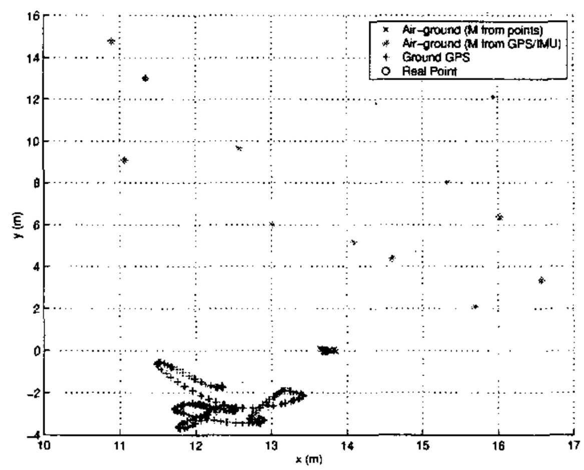


Fig. 8. GPS and air-ground localization of a fixed point.

TABLE I

POSITIONING ERRORS FOR GPS AND AIR-GROUND LOCALIZATION.

	Localization Error	Std. Deviation
Air-ground (M from points)	0.081 m	0.040 m
Air-ground (M from GPS/IMU)	8.147 m	3.882 m
Robot GPS	3.016 m	0.584 m

The errors vary significatively depending on which method is being used. The air-ground localization with M computed from image points is the most accurate, with less than $0.1m$ error. But, as mentioned before, it needs the position of 6 pairs of known world-image points which can be unfeasible in practical situations. The localization based solely on the robot GPS has an average error of about $3m$. This is close to the expected error for the type of GPS being used, even more if we consider that we have GPS errors both on the localization of the frame's origin and the robot position. The errors for the air-ground localization based on the blimp's GPS/IMU are very high (more than $8m$ error). In fact, this large error is caused by the composition of several small errors. First of all, we can expect the same GPS errors on the localization of the blimp. Also, small errors on the blimp attitude computed from the IMU can cause large errors when points $20m$ distant on the ground are projected into the image. Finally, as mentioned, we do not have a precise measurement of the blimp height which adds more errors in the localization. Just as an example, the world coordinates (X^w, Y^w) of a certain pixel in one image change from $[-0.560, -2.7940]$ to $[3.846, 1.686]$ if a 5^o rotation and a $3m$ translation are applied in each axis of the camera frame. So, there is a $6.284m$ error in the global localization when small errors are applied on the rotation and translation of the camera.

B. Estimating Robot Position

Using the same three methodologies, we estimated the position of a robot moving on the ground. The robot recorded its GPS coordinates while the blimp was flying over and capturing images. The black squares in Figure 7 highlight the robot positions on the images. Basically, the robot came from the top on the 45 yard line, turned to its right on the hash marks and turned right again on the 35 yard line. The graphs of Figure 9 show the robot trajectories computed using the two types of air-ground localization and the robot's GPS. In this run, the robot was remote controlled with a joystick, so we do not have a precise ground truth to compute the trajectory errors. But we can visually estimate from images 9 and 10 in Figure 7 that the robot crossed the 35 yard line hash mark. This point is shown with a black circle in the first graph of Figure 9.

The graphs show that the results obtained for localizing a moving target are similar to the ones obtained for a fixed point: air-ground localization with M computed from image points is very precise, the localization using ground GPS has a displacement of about $4m$ and the air-ground localization using blimp's GPS/IMU has larger errors. The path computed by the first method passes through the ground truth point and the one estimated from the ground GPS, in spite of the displacement, has a similar shape. But the errors in the air-ground localization using blimp's GPS/IMU make it impossible to track the correct robot path.

C. Discussion

The results demonstrate that none of these approaches could be applied alone if we need a localization system that is applicable, reliable, and accurate. In spite of being very precise, the air-ground localization using known image features can not always be applied. As explained in section IV-B, it requires the identification of a certain number of world locations to register the image, which is impractical in some situations. On the other hand, our GPS solution provided localization estimates within a $3m$ range, which may not be sufficient depending on the application. Moreover, we can not guarantee that GPS coverage will be available at all times, mainly when the robots are navigating in cluttered urban environments as described in Section IV-A. Finally, the approach based on the blimp's on board sensors (GPS/IMU) did not performed well. The combination of different sensor errors with no adequate

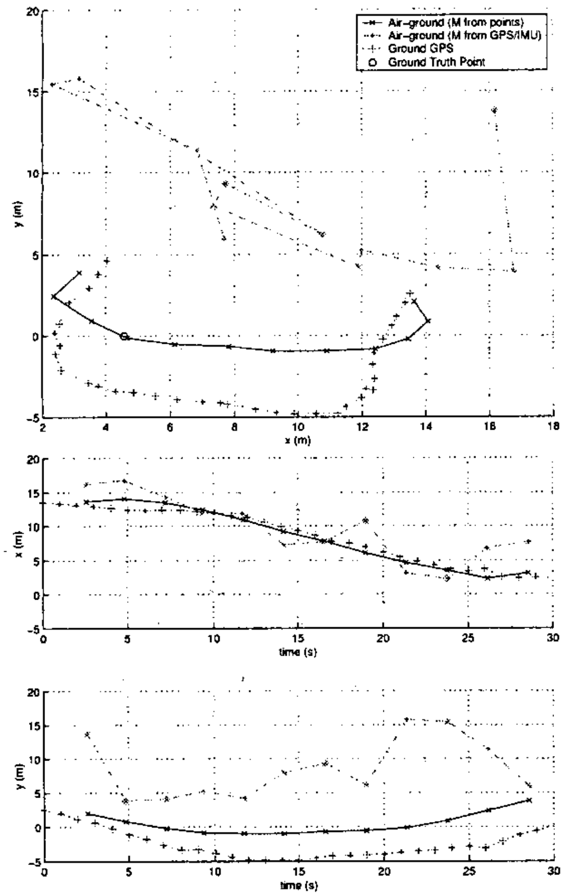


Fig. 9. Estimated robot trajectory.

fusion made this approach inappropriate for localizing ground robots within a desired error.

We believe that the best solution for having a reliable and accurate localization system is to have a mix of these approaches, fusing the information from the available sensors whenever necessary. For example, the blimp could improve its GPS and IMU estimates using some visual information returned from the camera. In this case, maybe a single known image point or the motion analysis of a sequence of images would suffice for a better estimation. More than that, combining information from different robots would certainly help the localization process. If two robots have an estimate of their position within a known error margin and can see each other, they can exchange information and reduce the overall uncertainty of their localization, especially in relative coordinates. For this, while executing their individual tasks, robots must be aware of their teammates in order to maintain network connectivity and cope with sensor range constraints. Consequently, the coordination of air and ground vehicles is a key factor for making this type of approach effective.

VI. Conclusion

In this paper we presented some experiments in multi-robot air-ground coordination. We introduced our hardware and software testbeds and analyzed different approaches for localizing ground robots using different sensors. The main conclusion is that individual approaches may not be sufficient for having a reliable and applicable localization system. In most cases, in order to reduce uncertainty, robots need to exchange information and combine estimates. For this, an adequate coordination mechanism is fundamental.

Our future work is directed towards several fronts. Firstly, we want to have the air-ground localization running online so that the ground robots will be able to exchange information with the blimp and have a better estimate of their global localization. For that the blimp will need to dynamically detect features on the ground and automatically register them to some world coordinate frame. One possible approach is to have a referenced aerial mosaic of the area built beforehand in a previous flight and then register the images captured from the blimp to this referenced frame. Also, we would like to do a reverse procedure and estimate the blimp's rotation and translation from the images and information that it receives from the ground robots. All this effort is part of a larger project with other research groups (University of Southern California, Georgia Tech, and BBN Technologies), in which heterogeneous multi-robot teams will be coordinated to perform a more sophisticated sensor fusion, dynamically gathering information from different local and remote sensors in order to have high level of situational awareness.

Acknowledgment

This work was in part supported by: DARPA MARS NBCH1020012, ARO MURI DAAD19-02-01-0383, and NSF CCR02-05336. The authors would like to thank the other members of the MARS group at the GRASP Laboratory for their help with the experiments and A. Bhusnurmath and R. Swaminathan for their help with the camera calibration.

References

[1] A. Elfes, M. Bergerman, J. Carvalho, E. Paiva, J. Ramos, and S. Bueno, "Air-ground robotic ensembles for cooperative applications: concepts and preliminary results," in *Proceedings of the International Conference on Field and Service Robotics*, 1999, pp. 75–80.

[2] S. Lacroix, I.-K. Jung, A. Mallet, and R. Chatila, "Towards cooperative air/ground robotics: issues related to environment modeling," in *10th International Conference on Advanced Robotics*, Budapest (Hungary), 2001.

[3] A. Stentz, T.and Kelly, H. Herman, P. Rander, and M. R., "Integrated air/ground vehicle system for semi-autonomous off-road navigation," in *Proceedings of AUVSIs Symposium on Unmanned Systems*, 2002.

[4] G. Sukhatme, J. Montgomery, and R. Vaughan, "Experiments with aerial-ground robots," in *Robot Teams: From Diversity to Polymorphism*, T. Balch and L. Parker, Eds. AK Peters, 2001.

[5] S. Sukkarieh, B. Grocholsky, E. Nettleton, and H. Durrant-Whyte, "Information fusion and control for multiple UAVs," in *NRL Workshop on Multi-Robot Systems*, vol. 2. Kluwer, March 2003.

[6] L. Chaimowicz, A. Cowley, V. Sabella, and C. J. Taylor, "ROCI: A distributed framework for multi-robot perception and control," in *Proceedings of the 2003 IEEE/RJS International Conference on Intelligent Robots and Systems*, 2003, pp. 266–271.

[7] E. Trucco and A. Verri, *Introductory Techniques for 3-D computer Vision*. Prentice Hall, 1998.